Bulletin of the JSME Vol.19, No.1, 2025
Journal of Advanced Mechanical Design, Systems, and Manufacturing

Trajectory tracking control of slidable-wheel omnidirectional mobile robot based on linear model predictive control

Huang XU*, Tatsuro TERAKAWA* and Masaharu KOMORI*
* Department of Mechanical Engineering and Science, Kyoto University
Kyoto daigaku-katsura, Nishikyo-ku, Kyoto 615-8540, Japan
E-mail: terakawa@me.kyoto-u.ac.jp

Received: 17 October 2024; Revised: 28 December 2024; Accepted: 6 March 2025

Abstract
Omnidirectional mobile robots with conventional wheels avoid the drawbacks (e.g., shock, slippage, and low load capacity) of omnidirectional mobile robots with wheels that have special structures. We previously proposed such a robot, called the slidable-wheel omnidirectional mobile robot (SWOM), as well as its controller for point-to-point movement. However, for practical applications, such as transporting goods in factories and warehouses, SWOM needs to be able to follow a predefined trajectory. In this paper, we present the design of a trajectory tracking controller for SWOM. Given that SWOM is a nonlinear system with constraints on both inputs and outputs, model predictive control (MPC) is adopted. Due to the high computational demands and time consumption associated with nonlinear MPC, linear MPC is used to achieve trajectory tracking. By expanding the previous research, an original method for generating a reference path that includes not only state variables but also inputs is proposed in this paper for the trajectory tracking task. The linearized kinematic model of SWOM is obtained using a first-order Taylor expansion around reference points on the reference path. Simulations considering slippage are conducted and the results show that SWOM can well track the reference path. Experiments conducted on a prototype also validate the effectiveness of the proposed control method.

Keywords: Model predictive control, Omnidirectional mobile robot, Trajectory tracking, Linearization, Reference path generation

1. Introduction

Omnidirectional mobile robots, which can move in any direction instantaneously, have become increasingly important in the field of mobile robotics. Omnidirectional movement can be achieved using specialized wheels, such as caster wheels (Lin et al., 2024; Kato and Wada, 2020; Medina and Hacohen, 2021), spherical wheels (Navabi et al., 2017; Zarei et al., 2018), and Mecanum wheels (Liu et al., 2017; Terakawa et al., 2023). However, the performance of robots that use such wheels is limited by the wheel structure. For example, caster wheels rotate around the steering axis with a small pause (Taheri and Zhao, 2020), which could cause shock or slippage, spherical wheels often have a relatively large volume, and Mecanum wheels tend to have a relatively low load capacity. To avoid these problems, the present authors previously proposed an omnidirectional mobile robot that uses conventional wheels, called the slidable-wheel omnidirectional mobile robot (SWOM) (Terakawa et al., 2018), as shown in Fig. 1.

Paper No.24-00370
[DOI: 10.1299/jamdsm.2025jamdsm0011]

© 2025 The Japan Society of Mechanical Engineers. This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs license (https://creativecommons.org/licenses/by-nc-nd/4.0/).

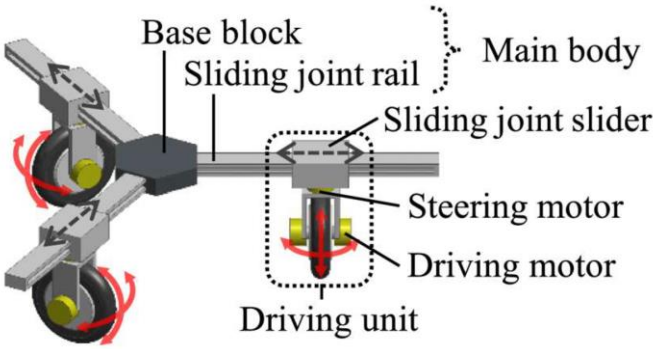


Fig. 1 Diagram of slidable-wheel omnidirectional mobile robot (SWOM) (Terakawa et al., 2018).

SWOM has a base block connected by three sliding rails. Each sliding rail has a driving unit connected to it via a passive sliding joint, which consists of a wheel and two motors. By using conventional wheels, SWOM avoids the drawbacks mentioned above. Previous research studied the kinematics and singularity of SWOM. A controller for the point-to-point movement of SWOM was also designed (Terakawa et al., 2019). However, further research is required before SWOM can be applied for practical applications. The next step is to develop trajectory tracking control, which would enable the robot to move along a pre-planned path designated by the user.

Trajectory tracking control has been extensively studied (Amoozgar et al., 2012; Li et al., 2016; Wang et al., 2014). Sliding mode control has been applied to trajectory tracking in various fields, including underwater vehicles (Qiao and Zhang, 2019, 2020), flapping-wing micro aerial vehicles (He et al., 2021), and mobile robots (Wendemagegn et al., 2024). A trajectory tracking control method for automated guided vehicles based on the Udwadia-Kalaba approach was proposed by Yu et al. (2024). Trajectory tracking from the perspective of port-controlled Hamiltonian systems was discussed by Fujimoto et al. (2003). Model predictive control (MPC) is commonly used for trajectory tracking (Neunert et al., 2016; Shen et al., 2018; Wang et al., 2018). The fundamental concept of MPC is to evaluate the system's future dynamics and minimize a cost function determined by the input. Due to its conceptual simplicity and ability to handle complex systems with strict control constraints and multiple inputs and outputs (Mayne, 2014), MPC is widely used in various industries, such as process control (Steyn and Sandrock, 2013), power electronics (Dragicevic and Novak, 2019), and manufacturing (Liu and Zhang, 2014; Wehr et al., 2020). MPC can be categorized into nonlinear MPC and linear MPC (LMPC) depending on the presence or absence of nonlinear terms, respectively. Compared to LMPC, nonlinear MPC requires greater computational effort as its optimization problems are often nonlinear and must be solved in real time. In practice, the nonlinear components of a problem can be linearized, allowing LMPC to be applied to problems involving nonlinearities.

Due to its low computational complexity, ease of implementation, and ability to handle constraints on inputs and outputs, LMPC was adopted to achieve trajectory tracking for SWOM. Building on the kinematic equations of SWOM proposed in previous research, this study introduces a novel method for generating reference trajectories. The generated reference trajectories encompass not only all state variables of SWOM but also the corresponding reference inputs. Subsequently, the linear motion equations of SWOM were derived through a first-order Taylor expansion at the reference points along the trajectory. Finally, the LMPC method was successfully applied to achieve trajectory tracking for SWOM, and its effectiveness was validated through both simulations and prototype experiments.

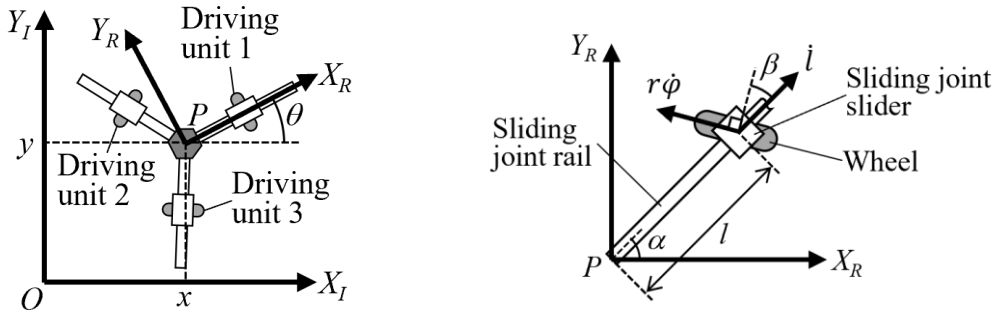
The rest of this paper is organized as follows. Section 2 reviews the kinematic model and singularity for SWOM. Section 3 presents the proposed methodology for generating reference trajectories and motion planning. Section 4 describes the linearization of the kinematic equations for SWOM and the application of LMPC to the trajectory tracking problem. A numerical simulation is conducted to verify the effectiveness of the proposed method. Section 5 presents and discusses the experimental results.

2. Kinematic analysis of slidable-wheel omnidirectional mobile robot

The global reference frame and local reference frame are shown in Fig. 2(a). The origin of the local reference frame is fixed at the center of the base block, with the X_R axis parallel to a sliding rail. To distinguish the three driving units, the unit that slides on the rail parallel to the X_R axis is labeled as driving unit 1. The other two driving units are

[DOI: 10.1299/jamdsm.2025jamdsm0011]
© 2025 The Japan Society of Mechanical Engineers

labeled as driving units 2 and 3, respectively, in the counterclockwise direction. Figure 2(b) shows more detailed parameters describing the SWOM's configuration in the local reference frame. The angle between each sliding rail and the X_R axis is denoted as α. l is the displacement of the slider, r is the wheels' radius, $\dot{\varphi}$ is the rotation speed of the wheel, and β is the angle between the wheel axle and the sliding rail indicating the orientation of the wheel. We use the subscript i to indicate the parameters belonging to driving unit i $(i = 1, 2, 3)$.



(a) (b)

Fig. 2 (a) Global and local reference frames for SWOM. (b) Wheel parameters in the local reference frame (Terakawa et al., 2019).

The kinematic model of SWOM is derived as:

$$[\sin(\alpha_i + \beta_i) \quad -\cos(\alpha_i + \beta_i) \quad -l_i\cos\beta_i]R(\theta)\begin{bmatrix}\dot{X}\\ \dot{Y}\\ \dot{\theta}\end{bmatrix} + r\dot{\varphi}_i + \dot{l}_i\sin\beta_i = 0 \tag{1}$$

$$[\cos(\alpha_i + \beta_i) \quad \sin(\alpha_i + \beta_i) \quad l_i\sin\beta_i]R(\theta)\begin{bmatrix}\dot{X}\\ \dot{Y}\\ \dot{\theta}\end{bmatrix} + \dot{l}_i\cos\beta_i = 0 \tag{2}$$

where

$$R(\theta) = \begin{bmatrix}\cos\theta & \sin\theta & 0\\ -\sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix} \tag{3}$$

and $\dot{X}$ represents the velocity of SWOM along the OX_I axis in the global coordinate system, while $\dot{Y}$ represents its velocity along the OY_I axis in the same system.

These equations are derived under the assumption of no-slip conditions, based on the relationship between the SWOM's body velocity and the velocities of its three wheels. Equation (1) describes the scenario where, under no-slip conditions, the actual rotational speed of the wheel equals the velocity component of the point on the sliding rail coinciding with the wheel in the local coordinate system, projected in the direction parallel to the wheel orientation, plus the displacement velocity of the wheel along the sliding rail in the same direction. Under no-slip conditions, the wheel's velocity component perpendicular to its orientation is zero. Accordingly, Eq. (2) states that the velocity component of the point on the sliding rail coinciding with the wheel in the local coordinate system, projected in the direction perpendicular to the wheel orientation, plus the displacement velocity of the wheel along the sliding rail in the same perpendicular direction, must sum to zero.

By combining Eqs. (1) and (2), we can eliminate $\dot{l}$. The forward and inverse kinematics can be expressed by Eqs. (4) and (5), respectively.

[DOI: 10.1299/jamdsm.2025jamdsm0011]
© 2025 The Japan Society of Mechanical Engineers

$$\begin{bmatrix}\dot{X}\\ \dot{Y}\\ \dot{\theta}\end{bmatrix} = R(\theta)^{-1}\begin{bmatrix}-\sin\alpha_1 & \cos\alpha_1 & l_1\\ -\sin\alpha_2 & \cos\alpha_2 & l_2\\ -\sin\alpha_3 & \cos\alpha_3 & l_3\end{bmatrix}^{-1}\begin{bmatrix}r\cos\beta_1 & 0 & 0\\ 0 & r\cos\beta_2 & 0\\ 0 & 0 & r\cos\beta_3\end{bmatrix}\begin{bmatrix}\dot{\varphi}_1\\ \dot{\varphi}_2\\ \dot{\varphi}_3\end{bmatrix} \tag{4}$$

$$\begin{bmatrix}\dot{\varphi}_1\\ \dot{\varphi}_2\\ \dot{\varphi}_3\end{bmatrix} = \begin{bmatrix}r\cos\beta_1 & 0 & 0\\ 0 & r\cos\beta_2 & 0\\ 0 & 0 & r\cos\beta_3\end{bmatrix}^{-1}\begin{bmatrix}-\sin\alpha_1 & \cos\alpha_1 & l_1\\ -\sin\alpha_2 & \cos\alpha_2 & l_2\\ -\sin\alpha_3 & \cos\alpha_3 & l_3\end{bmatrix}R(\theta)\begin{bmatrix}\dot{X}\\ \dot{Y}\\ \dot{\theta}\end{bmatrix} \tag{5}$$

The robot will fall into a singular configuration if the orientation of any of the wheels is parallel to the sliding rail. In this paper, we constrain β to the range of $(-\pi/2, \pi/2)$ to avoid the singularity. Substituting $\alpha_1 = 0, \alpha_2 = 2\pi/3$, and $\alpha_3 = 4\pi/3$ into the inverse kinematic equation yields

$$\begin{bmatrix}\dot{\varphi}_1\\ \dot{\varphi}_2\\ \dot{\varphi}_3\end{bmatrix} = \begin{bmatrix}r\cos\beta_1 & 0 & 0\\ 0 & r\cos\beta_2 & 0\\ 0 & 0 & r\cos\beta_3\end{bmatrix}^{-1} * \begin{bmatrix}-\dot{X}\sin\theta + \dot{Y}\cos\theta + \dot{\theta}l_1\\ \dot{X}\left(-\frac{\sqrt{3}}{2}\cos\theta + \frac{1}{2}\sin\theta\right) + \dot{Y}\left(-\frac{\sqrt{3}}{2}\sin\theta - \frac{1}{2}\cos\theta\right) + \dot{\theta}l_2\\ \dot{X}\left(\frac{\sqrt{3}}{2}\cos\theta + \frac{1}{2}\sin\theta\right) + \dot{Y}\left(\frac{\sqrt{3}}{2}\sin\theta - \frac{1}{2}\cos\theta\right) + \dot{\theta}l_3\end{bmatrix} \tag{6}$$

Due to the limitations of the motors and the restricted length of the sliding rail, SWOM has constraints on both inputs and outputs. We can satisfy the constraints on the inputs by directly controlling their magnitude. From Eq. (2), when $-\pi/2 < \beta_i < \pi/2$, the time dependence of l_i can be expressed as:

$$\dot{l}_i = -\dot{X}\big(\cos(\theta+\alpha_i) - \tan\beta_i\sin(\theta+\alpha_i)\big) - \dot{Y}\big(\tan\beta_i\cos(\theta+\alpha_i) + \sin(\theta+\alpha_i)\big) - l_i\dot{\theta}\tan\beta_i \tag{7}$$

From Eq. (7), it is possible to keep l_i within the constraint limits by controlling the orientation of the wheel.

3. Reference path generation

Since the kinematic system is linearized using a first-order Taylor expansion at reference points on the reference path, we need a reference trajectory for SWOM that includes not only its configuration but also the corresponding inputs. We first define the target velocity for SWOM and then calculate the reference value for the wheels' rotational speed. Based on this, the orientation of each wheel is adjusted to ensure that l_i remains within the specified range. Consequently, reference values for l_i, β_i, and $\dot{\beta}_i$ are obtained. The procedures for reference path generation are as follows.

The inputs of the system are the rotation and steering velocity of the wheels, $\dot{\varphi}_i$ and $\dot{\beta}_i$, which means we cannot directly control the steering angle of the wheel. At the k-th time step, the reference translation velocity $(\dot{X}_{ref}(k), \dot{Y}_{ref}(k))$ and rotation velocity $\dot{\theta}_{\mathrm{ref}}(k)$ of SWOM are first specified. The configurations of SWOM, including $X(k), Y(k), \theta(k), l_i(k)$, and $\beta_i(k)$, are measured by sensors. We can then calculate the necessary rotation velocity of the three wheels $\dot{\varphi}_{id}(k)$ using Eq. (6).

Next, the reference steering velocity of the wheels $\dot{\beta}_{i\mathrm{ref}}(k)$ is designed to stabilize the displacement of the slider. Let $E_{li} = l_{iref} - l_{id}$ be the distance between the slider's position and the desired position, where l_{id} denotes the desired position of the slider. Generally, it is set to be at the middle of the sliding rail. If $\dot{l}_{i\mathrm{ref}} = -K_i E_{li}$, $l_{i\mathrm{ref}}$ will converge to l_{id}. With $\dot{l}_{i\mathrm{ref}} = -K_i E_{li}$, from Eq. (7), we can obtain the reference orientation of the wheel $\beta_{i\mathrm{ref}}$ as:

$$\beta_{i\mathrm{ref}} = \arctan\left(\frac{\dot{X}_{\mathrm{ref}}\cos(\theta+\alpha_i) + \dot{Y}_{\mathrm{ref}}\sin(\theta+\alpha_i) - K_i E_{li}}{\dot{X}_{\mathrm{ref}}\sin(\theta+\alpha_i) - \dot{Y}_{\mathrm{ref}}\cos(\theta+\alpha_i) - l_i\dot{\theta}_{\mathrm{ref}}}\right) \tag{8}$$

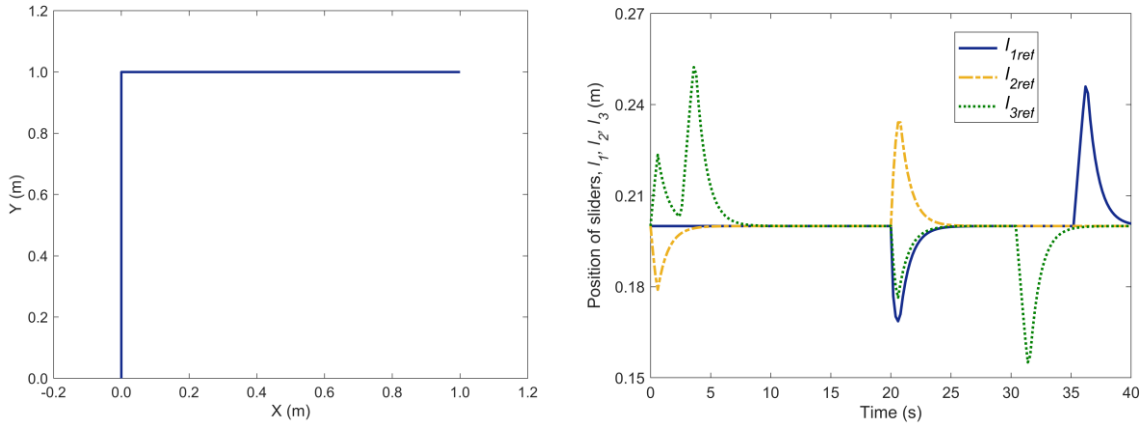
At the k-th time step, the rate of change of the position of the slider $\dot{l}_{i\mathrm{ref}}(k)$ is first derived according to Eq. (7). Then, the position of the slider at the next time step $l_{i\mathrm{ref}}(k+1)$ and the distance between the slider and its desired position at the next time step $E_{li}(k+1)$ can be predicted. Using Eq. (8), the reference steering angle of the wheel for the next time step $\beta_{\mathrm{ref}}(k+1)$ is derived to control the displacement of the slider. Thus, the reference angular velocity

[DOI: 10.1299/jamdsm.2025jamdsm0011]
© 2025 The Japan Society of Mechanical Engineers

$\dot{\beta}_{i\mathrm{ref}}(k)$ can be calculated from the rate of change of the steering angle between two steps to ensure the angular velocity at the next step reaches the desired value (i.e., $\dot{\beta}_{i\mathrm{ref}}(k) = (\beta_{i\mathrm{ref}}(k+1) - \beta_{i\mathrm{ref}}(k))/dt$). If $\dot{X}_{\mathrm{ref}}\sin(\theta+\alpha_i) - \dot{Y}_{\mathrm{ref}}\cos(\theta+\alpha_i) - l_i\dot{\theta}_{\mathrm{ref}} = 0$, $\beta_{i\mathrm{ref}}(k+1)$ becomes $\pi/2$ or $-\pi/2$, which violates the constraints placed on the steering angle of the wheel. For this situation, we simply set $\beta_{i\mathrm{ref}}(k+1)$ to the value at the previous step and $\dot{\beta}_{i\mathrm{ref}}(k) = 0$.

The reference position for SWOM can be calculated by integrating $\dot{X}_{\mathrm{ref}}, \dot{Y}_{\mathrm{ref}}$, and $\dot{\theta}_{\mathrm{ref}}$. Finally, we obtain the reference path for SWOM, which consists of the reference configuration and the corresponding inputs.

An example of a reference trajectory and the corresponding reference position of the sliders are shown in Fig. 3. To demonstrate the robot's omnidirectional mobility, the reference trajectories are set to be unsmooth and non-differentiable. The target velocity of the robot is ($\dot{X} = 0.05\,\mathrm{m/s}, \dot{Y} = 0.00\,\mathrm{m/s}$) for the first 20.0 s and ($\dot{X} = 0.00\,\mathrm{m/s}, \dot{Y} = 0.05\,\mathrm{m/s}$) for the next 20.0 s. The target rotation speed is 0.08 rad/s. The target position of the sliders is set to $l_{id} = 0.20$ m and the limits on the position of the sliders are $l_{\min} = 0.12$ m and $l_{\max} = 0.30$ m. K_i ($i = 1, 2, 3$) is set as 1. In the subsequent simulations and experiments, the value of K_i has also been set to 1. Figure 3(b) shows the reference position of the sliders as the robot moves along the reference trajectory. From the graph, we can conclude that the position of the sliders can be limited to a small range from the specified position using the proposed method.



(a) (b)

Fig. 3 (a) Reference path and (b) reference position for sliders on sliding rails.

The graph of the position of the sliders has some peaks. There are two main reasons for this phenomenon. First, when the velocity of the robot suddenly changes a lot, the orientation of the wheels β also needs to change a lot. Since we set an upper limit on the acceleration of β, the desired value is not reached immediately, so the displacement of the slider continues to change. Second, the direction of the velocity at the point on the robot that is in contact with an arbitrary slider is parallel to the direction of its sliding rail (i.e., $\dot{X}_{\mathrm{ref}}\sin(\theta+\alpha_i) - \dot{Y}_{\mathrm{ref}}\cos(\theta+\alpha_i) - l_i\dot{\theta}_{\mathrm{ref}} = 0$). At this point, β_i needs to be $\pi/2$ or $-\pi/2$, which is prohibited, and Eq. (7) becomes:

$$\dot{l}_i = -\dot{X}_{\mathrm{ref}}\cos(\theta+\alpha_i) - \dot{Y}_{\mathrm{ref}}\sin(\theta+\alpha_i) \tag{9}$$

The rate of change of the displacement of the slider $\dot{l}_i$ equals the velocity of the robot in the direction of the sliding rail. Therefore, the displacement of the slider continues to change and the speed becomes the same as that of the robot in the direction of the sliding rail.

4. Trajectory tracking for SWOM

4.1 Linearization of kinematic equation

We first define the state variables and inputs for SWOM as:

[DOI: 10.1299/jamdsm.2025jamdsm0011]
© 2025 The Japan Society of Mechanical Engineers

$$q = [X, Y, \theta, l_1, l_2, l_3, \beta_1, \beta_2, \beta_3],$$
$$U = [\dot{\varphi}_i, \dot{\varphi}_2, \dot{\varphi}_3, \dot{\beta}_1, \dot{\beta}_2, \dot{\beta}_3] \tag{10}$$

The derivatives of X, Y, and θ can be calculated from U using Eq. (4). By substituting Eq. (4) into Eq. (2), we can calculate $[\dot{l}_1, \dot{l}_2, \dot{l}_3]$ from U as

$$\begin{bmatrix} \dot{l}_1 \\ \dot{l}_2 \\ \dot{l}_3 \end{bmatrix} = -\begin{bmatrix} \cos\beta_1 & 0 & 0 \\ 0 & \cos\beta_2 & 0 \\ 0 & 0 & \cos\beta_3 \end{bmatrix}^{-1} \begin{bmatrix} \cos\beta_1 & \sin\beta_1 & l_1\sin\beta_1 \\ \cos(\beta_2+\alpha_2) & \sin(\beta_2+\alpha_2) & l_2\sin\beta_2 \\ \cos(\beta_3+\alpha_3) & \sin(\beta_3+\alpha_3) & l_3\sin\beta_3 \end{bmatrix} * \begin{bmatrix} -\sin\alpha_1 & \cos\alpha_1 & l_1 \\ -\sin\alpha_2 & \cos\alpha_2 & l_2 \\ -\sin\alpha_3 & \cos\alpha_3 & l_3 \end{bmatrix}^{-1} \begin{bmatrix} r\cos\beta_1 & 0 & 0 \\ 0 & r\cos\beta_2 & 0 \\ 0 & 0 & r\cos\beta_3 \end{bmatrix} \begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \\ \dot{\varphi}_3 \end{bmatrix} \tag{11}$$

In addition, β_i can be calculated by integrating $\dot{\beta}_i$. Therefore, the derivative of q can be expressed as a function of q and U.

$$\dot{q} = f(q, U) \tag{12}$$

We discretize the system at the k-th time step as

$$q(k+1) = q(k) + \dot{q}(k)dt = q(k) + f(q(k), U(k))dt \tag{13}$$

The state variables at the next time step can be represented by a function of the state variables and inputs of the current time step.

$$q(k+1) = F\big(q(k), U(k)\big) = q(k) + f(q(k), U(k))dt \tag{14}$$

At the point on the reference path at time step $\big(q_d(k), U_d(k)\big)$, the system also satisfies

$$q_d(k+1) = F\big(q_d(k), U_d(k)\big) = q_d(k) + f\big(q_d(k), U_d(k)\big)dt \tag{15}$$

For $q(k), U(k)$ close to $q_d(k), U_d(k)$, from Taylor's theorem, $F\big(q(k), U(k)\big)$ can be calculated as

$$\begin{aligned} q(k+1) &= F(q(k), U(k)) \\ &= F\big(q_d(k), U_d(k)\big) + \frac{\partial F}{\partial q}\Big|_{q=q_d(k)} * \big(q(k) - q_d(k)\big) + \frac{\partial F}{\partial U}\Big|_{U=U_d(k)} * \big(U(k) - U_d(k)\big) \end{aligned} \tag{16}$$

where $\partial F/\partial q|_{q=q_d(k)}$ and $\partial F/\partial U|_{U=U_d(k)}$ are the partial derivatives of the function $F\big(q(k), U(k)\big)$ at $\big(q_d(k), U_d(k)\big)$ with respect to q and U, respectively. The linear and discrete kinematic equations for SWOM are thus obtained.

4.2 Trajectory tracking control using model predictive control

In Eq. (16), $F\big(q_d(k), U_d(k)\big) = q_d(k+1)$. We define the state error as $e^q(k) = q(k) - q_d(k)$ and the input error as $e^U(k) = U(k) - U_d(k)$. Then,

[DOI: 10.1299/jamdsm.2025jamdsm0011]
© 2025 The Japan Society of Mechanical Engineers

$$e^q(k+1) = \left.\frac{\partial F}{\partial q}\right|_{q=q_d(k)} e^q(k) + \left.\frac{\partial F}{\partial U}\right|_{U=U_d(k)} e^U(k) \tag{17}$$

$e^q(k+1)$ is the predicted state error of the $(k+1)$-th time step at the k-th time step, so we denote it as $e^q(k+1|k)$. $\partial F/\partial q|_{q=q_d(k)}$ and $\partial F/\partial U|_{U=U_d(k)}$, which are the partial derivatives of the function $F\big(q(k), U(k)\big)$ at $\big(q_d(k), U_d(k)\big)$, can be rewritten as $A(k)$ and $B(k)$, respectively.

The state error for the next N time steps can be calculated using Eq. (17), giving

$$\begin{aligned}
e^q(k+1|k) &= A(k)e^q(k) + B(k)e^U(k)\\
e^q(k+2|k) &= A(k+1|k)e^q(k+1|k) + B(k+1|k)e^U(k+1)\\
&= A(k+1|k)\,A(k)e^q(k) + A(k+1|k)B(k)e^U(k) + B(k+1|k)e^U(k+1)\\
&\vdots\\
e^q(k+N|k) &= A(k+N-1|k)A(k+N-2|k)\dots A(k)e^q(k)\\
&\quad +(k+N-1|k)A(k+N-2|k)\dots A(k+1|k)B(k)e^U(k)\\
&\quad +(k+N-1|k)A(k+N-2|k)\dots A(k+2|k)B(k+1|k)e^U(k+1)\\
&\quad +\cdots\\
&\quad +B(k+N-1|k)e^U(k+N-1)
\end{aligned} \tag{18}$$

where $e^q(k+N|k)$ is the predicted state error of the $(k+N)$-th time step at the k-th time step, $e^U(k+N-1)$ is the input error at the $(k+N-1)$-th time step, and $A(k+N-1|k) = \partial F/\partial q|_{q=q_d(k+N-1)}$ and $B(k+N-1|k) = \partial F/\partial U|_{U=U_d(k+N-1)}$ are respectively the predicted partial derivatives of the function $F\big(q(k), U(k)\big)$ with respect to q and U at the k-th time step at $\big(q_d(k+N-1), U_d(k+N-1)\big)$.

At time step k, the predicted state and input error for the next N steps can be written as

$$\overline{X}(k+1) = \begin{bmatrix} e^q(k+1|k)\\ e^q(k+2|k)\\ e^q(k+3|k)\\ \vdots\\ e^q(k+N|k)\end{bmatrix}, \overline{U}(k) = \begin{bmatrix} e^U(k)\\ e^U(k+1)\\ e^U(k+2)\\ \vdots\\ e^U(k+N-1)\end{bmatrix} \tag{19}$$

Equation (16) becomes

$$\overline{X}(k+1) = \bar{A}e^q(k) + \bar{B}\overline{U}(k) \tag{20}$$

where

$$\bar{A} = \begin{bmatrix} A(k)\\ A(k+1|k)A(k)\\ \vdots\\ A(k+N-1|k)\dots A(k)\end{bmatrix} \tag{21}$$

$$\bar{B} = \begin{pmatrix} B(k) & \cdots & \cdots & 0\\ A(k+1|k)B(k) & B(k+1|k) & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ A(k+N-1|k)\dots A(k+1|k)B(k) & A(k+N-1|k)\dots A(k+2|k)B(k+1|k) & \cdots & B(k+N-1|k)\end{pmatrix} \tag{22}$$

[DOI: 10.1299/jamdsm.2025jamdsm0011]
© 2025 The Japan Society of Mechanical Engineers

Then, the cost function is defined as

$$\bar{J} = \overline{X}(k+1)^T \bar{Q} \overline{X}(k+1) + \overline{U}(k)^T \bar{R} \overline{U}(k) \tag{23}$$

where $\bar{Q}$ and $\bar{R}$ are weight coefficient matrices.

Substituting Eq. (17) into Eq. (19) yields

$$\begin{aligned} \bar{J} &= \left(\bar{A}e^q(k) + \bar{B}\overline{U}(k)\right)^T \bar{Q}\left(\bar{A}e^q(k) + \bar{B}\overline{U}(k)\right) + \overline{U}(k)^T \bar{R}\overline{U}(k) \\ &= \overline{U}(k)^T(\bar{B}^T\bar{Q}\bar{B} + \bar{R})\overline{U}(k) + e^q(k)^T\bar{A}^T\bar{Q}\bar{B}\overline{U}(k) + \overline{U}(k)^T\bar{B}^T\bar{Q}\bar{A}e^q(k) + e^q(k)^T\bar{A}^T\bar{Q}\bar{A}e^q(k) \\ &= \frac{1}{2}\overline{U}(k)^T H(k)\overline{U}(k) + 2f(k)^T\overline{U}(k) + d(k) \end{aligned} \tag{24}$$

where $H(k) = 2(\bar{B}^T\bar{Q}\bar{B} + \bar{R})$, $f(k)^T = e^q(k)^T\bar{A}^T\bar{Q}\bar{B}$, and $d(k) = e^q(k)^T\bar{A}^T\bar{Q}\bar{A}e^q(k)$.

The minimization of the cost function becomes a quadratic programming problem with constraints on the inputs and outputs:

$$\begin{aligned} &\min_{\overline{U}(k)} \bar{J} = \frac{1}{2}\overline{U}(k)^T H(k)\overline{U}(k) + 2f(k)^T\overline{U}(k) + d(k) \\ &\text{s.t.} \\ &l_{\min} \le l_i \le l_{\max}, \\ &-\frac{\pi}{2} < \beta_i < \frac{\pi}{2}, \\ &\dot{\varphi}_{\min} \le \dot{\varphi}_i \le \dot{\varphi}_{\max}, \\ &\dot{\beta}_{\min} \le \dot{\beta}_i \le \dot{\beta}_{\max},\ i \in \{1,2,3\} \end{aligned} \tag{25}$$

At the k-th time step, the quadratic programming problem is solved and the optimal input error $\overline{U}(k)$ is derived. We only use the optimal input for the present time step and then advance to the next step.

A simulation of SWOM tracking the reference path shown in Fig. 3 was conducted to verify the effectiveness of the LMPC-based trajectory tracking method. Both the acceleration and deceleration of the robot were considered. Initially, the robot started with $\dot{X}_d = 0.00\,\mathrm{m/s}$, $\dot{Y}_d = 0.00\,\mathrm{m/s}$, and $\dot{\theta}_d = 0.00\,\mathrm{rad/s}$, and accelerated at $\ddot{Y}_d = 0.01\,\mathrm{m/s^2}$ and $\ddot{\theta}_d = 0.02\,\mathrm{rad/s^2}$. After 5.0 s, the robot moved at a constant speed of $\dot{X}_d = 0.00\,\mathrm{m/s}$, $\dot{Y}_d = 0.05\,\mathrm{m/s}$, and $\dot{\theta}_d = 0.1\,\mathrm{rad/s}$. When the robot approached the corner, $\ddot{Y}_d$ became $-0.01\,\mathrm{m/s^2}$ and $\ddot{\theta}_d$ became $-0.02\,\mathrm{rad/s^2}$. After 4.0 s, $\ddot{\theta}_d$ became 0 and $\ddot{Y}_d$ remained at $-0.01\,\mathrm{m/s^2}$. The robot finally stopped at the corner after one more second and rotated at $\dot{\theta}_d = 0.02\,\mathrm{rad/s}$. After that, the robot accelerated again at $\ddot{X}_d = 0.01\,\mathrm{m/s^2}$ and $\ddot{\theta}_d = 0.02\,\mathrm{rad/s^2}$ one second later. The accelerated rotation lasted 4.0 s and the accelerated movement in the X direction lasted 5.0 s. Then, the robot started moving at a constant speed of $\dot{X}_d = 0.05\,\mathrm{m/s}$, $\dot{Y}_d = 0.00\,\mathrm{m/s}$, and $\dot{\theta}_d = 0.1\,\mathrm{rad/s}$ until the end of the simulation.

The reference position for the sliders was set to $l_{id} = 0.20\,\mathrm{m}$ and the limits on the position of the sliders were $l_{\min} = 0.12$ m and $l_{\max} = 0.30$ m. The constraints on the inputs were $\dot{\varphi}_{\min} = -1.00\,\mathrm{rad/s}$, $\dot{\varphi}_{\max} = 1.00\,\mathrm{rad/s}$, $\dot{\beta}_{\min} = -2.50\,\mathrm{rad/s}$, and $\dot{\beta}_{\max} = 2.50\,\mathrm{rad/s}$. In the optimization problem of MPC, the state weight matrix is designed as $\bar{Q} = \mathrm{diag}(1000, 1000, 500, 100, 100, 100, 1, 1, 1)$, and the control weight matrix is $\bar{R} = I_{3\times3}$.

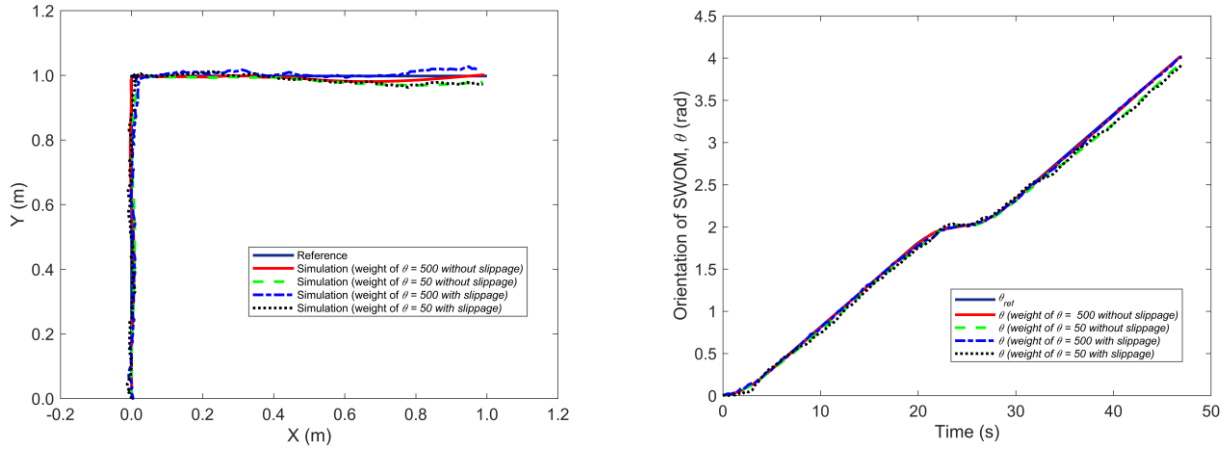
In the simulation, we simulate the effect of wheel slippage by adding random disturbances with an amplitude of 0.005 m to the position coordinates of the three wheels in the global coordinate system. Additionally, we use two different weights (500 and 50) for the errors related to θ to observe their impact on the simulation results. The tracking error of each simulation is recorded in Table. 1. Figure 4 presents the results of the four simulations, it can be observed that SWOM is able to track the reference path effectively in the simulation. In Fig. 4(b), it is evident that,

[DOI: 10.1299/jamdsm.2025jamdsm0011]
© 2025 The Japan Society of Mechanical Engineers

after 40 seconds, when the weights of the errors related to θ are the same, the difference between considering or not considering slipping is minimal. However, when the weight of the error related to θ is larger, SWOM is able to track the reference value of θ more effectively.

Table 1 Tracking error of simulations

Weight of θ = 500, without slippage	Mean squared error	0.0004
	Max position error	0.0385
	Max angle error	0.0201
Weight of θ = 50, without slippage	Mean squared error	0.0036
	Max position error	0.0361
	Max angle error	0.1156
Weight of θ = 500, with slippage	Mean squared error	0.0009
	Max position error	0.0572
	Max angle error	0.0502
Weight of θ = 50, with slippage	Mean squared error	0.0050
	Max position error	0.0417
	Max angle error	0.1421



(a) (b)

Fig. 4 (a) Simulated and reference paths and (b) simulated and reference orientations for SWOM.

Figure 5 shows the simulated position of the sliders as SWOM moved along the single-corner reference path. The tracking error is small and the simulation results remain in a small range from the reference position of the sliders. Compared to Fig. 3, there are fewer peaks in reference value of l_i because when we consider the acceleration and deceleration of the robot, there are no rapid changes in the velocity of the robot. The peaks only occur when the direction of the velocity at the point on the robot that is in contact with the slider is parallel to the direction of the sliding rail (i.e., when β_i needs to be $\pi/2$ or $-\pi/2$). Compared to the case with wheel slippage, when slippage is not considered, the position of the sliders changes more smoothly because there are no disturbances in the position of the wheels.

[DOI: 10.1299/jamdsm.2025jamdsm0011]
© 2025 The Japan Society of Mechanical Engineers

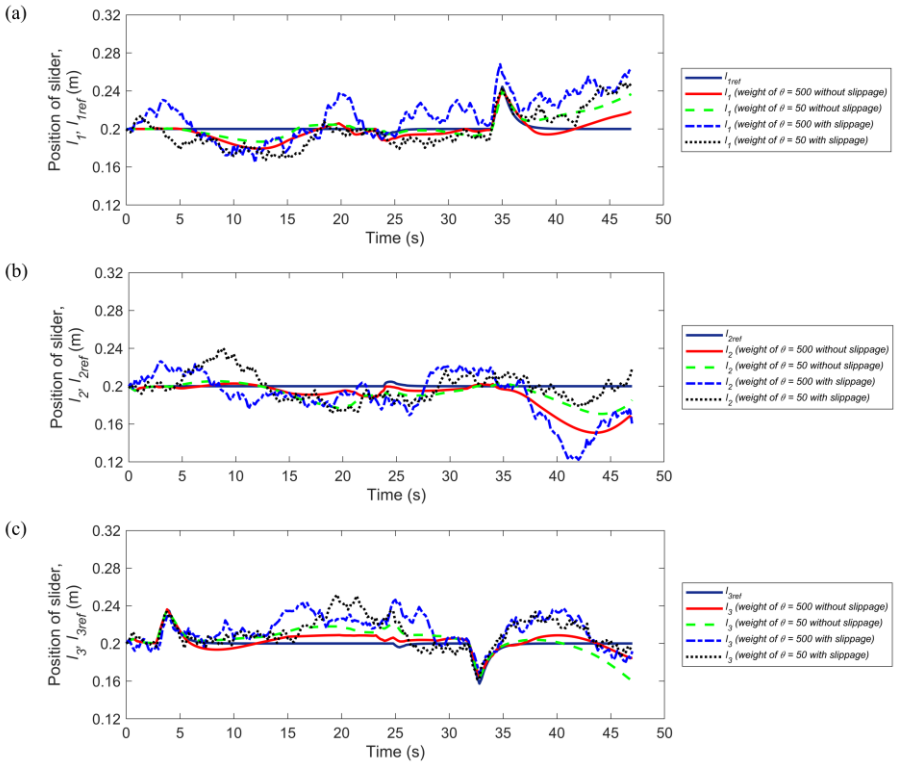


Fig. 5 Simulated and reference positions of sliders on SWOM.

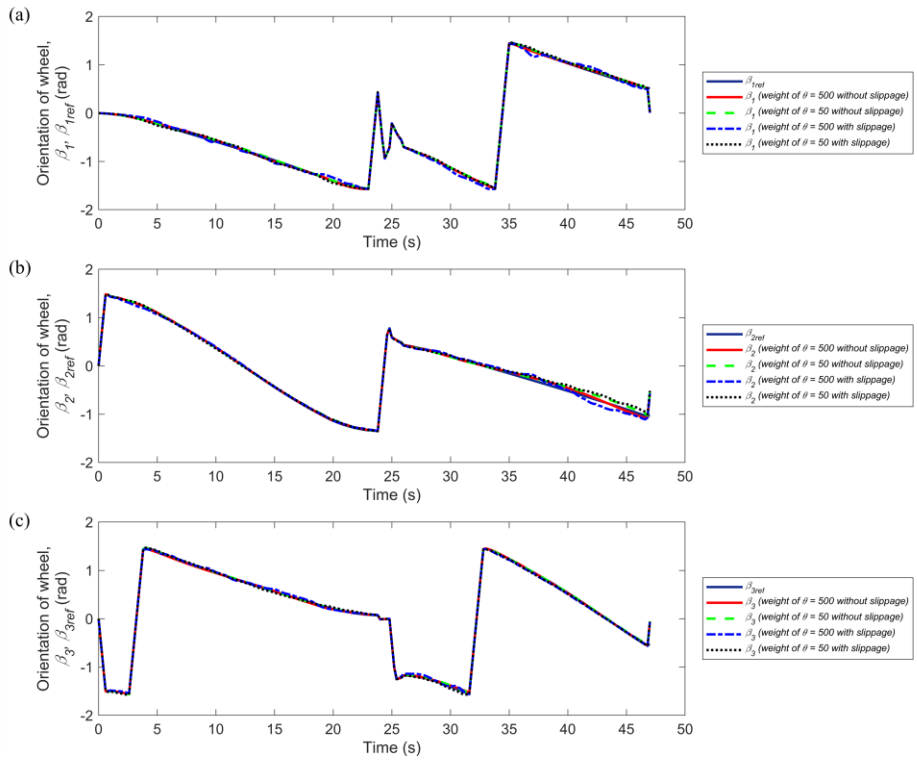


Fig. 6 Simulated and reference orientations of wheels on SWOM.

Figure 6 shows the orientation of the robot's wheels during the simulation compared to the reference values. The results show that the controller works well and that the wheels' orientations are almost the same as the reference values. It can also be seen that whether or not wheel slippage is considered, as well as the weight of the error related to θ, does not affect the tracking of the reference value for the wheel orientation.

[DOI: 10.1299/jamdsm.2025jamdsm0011]
© 2025 The Japan Society of Mechanical Engineers

5. Experiment

The previous section demonstrated that the proposed controller works well in a simulation. In this section, we conduct trajectory tracking experiments on a prototype to verify the effectiveness of the proposed controller.

5.1 Prototype of SWOM

The prototype, whose basic structure is similar to that in Fig. 1, is shown in Fig. 7. A robot computer and a monitor were placed on the top of the base block. Three sliding joints were connected to the base block. The angles between the joints were 120°. Each sliding joint consisted of two linear guides. Three driving units were attached to the three sliding joints, respectively. The units could move freely in the direction of the linear guides. A driving unit mainly consisted of two motors and a wheel. The upper motor was responsible for wheel steering and the lower motor was responsible for wheel rotation. In addition, each wheel was equipped with two ball rollers as safety wheels to prevent the robot from tipping over. These rollers did not affect the movement of SWOM because they rarely touched the ground.

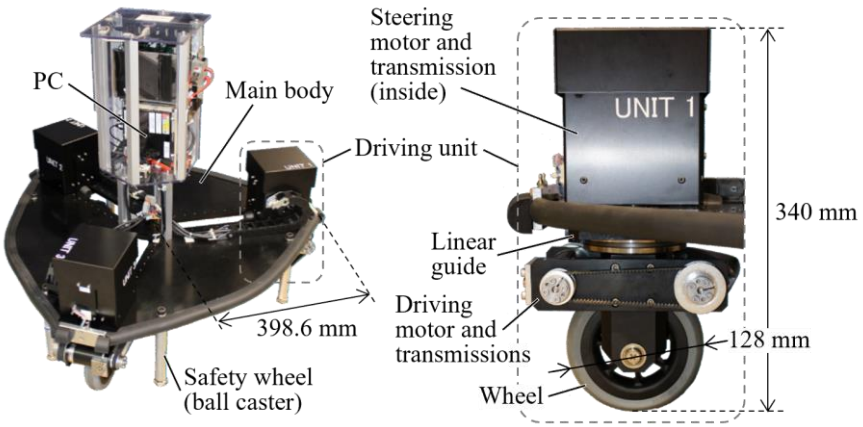


Fig. 7 Prototype of SWOM (Terakawa et al., 2019).

Figure 8 shows the control system of SWOM prototype. To measure the state variables for the robot to be used as inputs to the controller, a motion capture system (Bonita cameras, VICON) was used to measure the positions of the base block and the driving units and the displacement of the linear guides was calculated from the measured positions. The angle and speed of the wheels were measured by rotary encoders mounted on the motors. These measurements were sent to the robot computer, which then sent appropriate input commands to the motors via motor drivers. Details of the SWOM parameters are shown in Table 2.

[DOI: 10.1299/jamdsm.2025jamdsm0011]
© 2025 The Japan Society of Mechanical Engineers

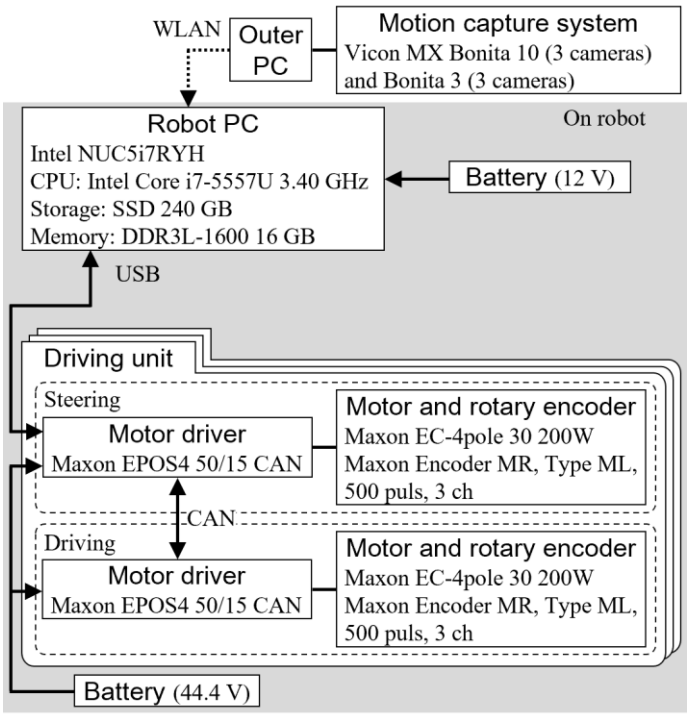


Fig. 8 Control system of SWOM prototype (Terakawa et al., 2019).

Table 2 Specifications of SWOM prototype (Terakawa et al., 2019).

Whole system	Weight	33kg
Driving unit (Driving / Steering)	Maximum speed	9.32 / 10.7 rad/s
	Maximum torque	38.0 / 41.7 N·m
	Resolution of encoder	1.35×10^{-5} / 1.54×10^{-5} rad
Linear guide	Movable range	$0.122\ \mathrm{m} \le l_i \le 0.314\ \mathrm{m}$
Motion capture	Operation period	1000 Hz

5.2 Experiments results

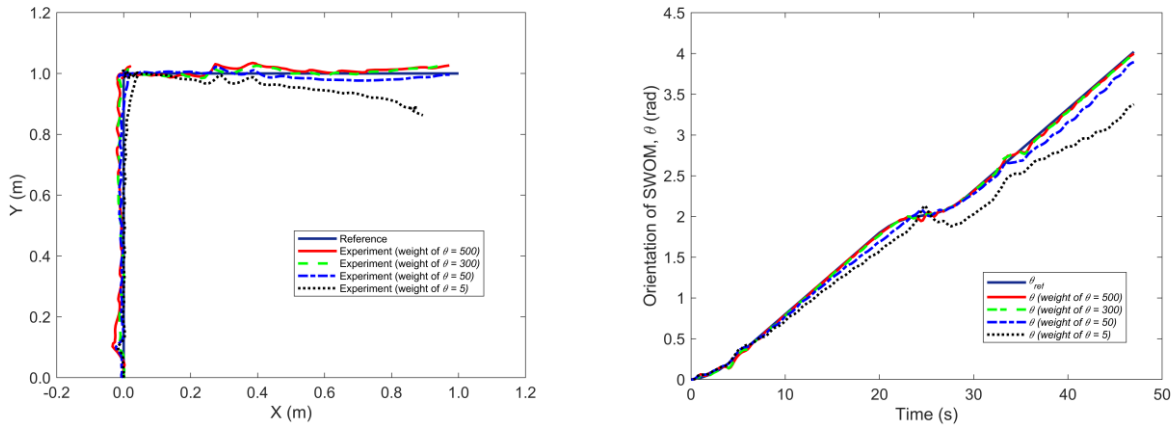
The prototype was tested on two different reference paths considering different weights of error of θ. The first path is a single-corner path, similar to that used in the simulation. Such a non-smooth path can only be traced by an omnidirectional mobile robot. The acceleration and deceleration of the robot were the same as those for the reference path in the simulation. The reference position of the sliders l_{ref} was set to 0.200 m and the limitation on the position of the sliders was $0.122\ \mathrm{m} \le l_i \le 0.314\ \mathrm{m}$. The constraints on the inputs were $\dot{\varphi}_{\min} = -1.00\ \mathrm{rad/s}$, $\dot{\varphi}_{\max} = 1.00\ \mathrm{rad/s}$, $\dot{\beta}_{\min} = -1.50\ \mathrm{rad/s}$, and $\dot{\beta}_{\max} = 1.50\ \mathrm{rad/s}$. The initial conditions were $X = 0.00$ m, $Y = 0.00$ m, $\theta = 0.00$ rad, and $\beta_1 = \beta_2 = \beta_3 = 0.00$ rad. The initial value of l_i was set randomly so that the slider was located near the middle of the slide rail.

Multiple experiments were conducted for different weights of the error of θ, with the tracking errors recorded in Table 3. When the weights of the error of θ are not too small, SWOM performs well in tracking the reference trajectory. Figure 9 shows the robot's path and the reference path during trial No. 2, as well as the robot's orientation compared to the reference orientation. From the figure, we can visually observe that when the weights of the error of θ are 500, 300, and 50, the tracking performance of the reference position is quite good. Moreover, when the weights of the error of θ are 500 and 300, the tracking of the robot's reference orientation is better compared to when the weights are 50 and 5. When the weight of the error of θ is 50 or 5, the orientation angle error may persist until the end. From the above, in this case, increasing the weights regarding the error of θ allows SWOM to better track the reference value of θ. Furthermore, when the weight of the error of θ becomes sufficiently large, further increasing the weight does not improve the tracking performance.

[DOI: 10.1299/jamdsm.2025jamdsm0011]
© 2025 The Japan Society of Mechanical Engineers

Table 3 Tracking error of each experiment (single-corner path).

		No. 1	No. 2	No. 3	No. 4	No. 5
Weight of $\theta = 500$	Mean squared error	0.0016	0.0015	0.0014	0.0019	0.0018
	Max position error	0.0565	0.0578	0.0502	0.0624	0.0663
	Max angle error	0.1020	0.0874	0.1031	0.1060	0.1132
Weight of $\theta = 300$	Mean squared error	0.0015	0.0013	0.0022	0.0015	0.0018
	Max position error	0.0518	0.0467	0.0687	0.0589	0.0627
	Max angle error	0.0880	0.0968	0.1084	0.0982	0.0955
Weight of $\theta = 50$	Mean squared error	0.0057	0.0087	0.0082	0.0112	0.0077
	Max position error	0.0395	0.0455	0.0437	0.0522	0.0398
	Max angle error	0.1336	0.1917	0.1633	0.1955	0.1778
Weight of $\theta = 5$	Mean squared error	0.0610	0.0911	0.1152		
	Max position error	0.1365	0.1738	0.1975		
	Max angle error	0.6021	0.6740	0.7273		



(a) (b)

Fig. 9 (a) Real and reference paths and (b) real and reference orientations of SWOM in single-corner path experiment.

Figure 10 shows the displacement of the sliders and the reference displacement during movement. The displacement of the sliders in the experiment did not match the reference value as closely as in the simulation, but it was consistently within the allowable range and remained close to the reference value. Figure 11 shows the orientation of the robot's wheels during the experiment compared to the reference values. As can be seen in Fig. 11, the experimental values oscillate around the reference values. This may be due to the controller's inability to precisely control the actual value of β. First, in the prototype, the wheel orientation can be measured using a rotary encoder; however, the zero point was set manually and might be affected by measurement errors or assembly errors. As a result, the measured value of β contains errors compared to the actual value. Secondly, when solving the optimization problem in the controller, we set the time step to 0.2 seconds, expecting the motor to rotate for 0.2 seconds at the specified speed each time it receives a command. However, in the experiment, the time step typically ranges from 0.187 seconds to 0.203 seconds, meaning that we cannot transmit commands with perfect accuracy every 0.2 seconds and have the motor run exactly for 0.2 seconds. Consequently, the actual value of β cannot reach the intended value accurately. These two factors lead to the controller's inability to precisely control the actual value of β, causing β to oscillate near the reference value to ensure that the other state variables of SWOM can track their reference values.

Comparing the results in Table 3 and Table 1, under no-slip conditions, when the weight of the error related to θ is the same, the maximum position error and maximum angle error in the simulation are both smaller than those in the experiment. When slippage is considered, with the same weight for the error related to θ, the maximum position and angle errors in the simulation are sometimes comparable to the experimental results. However, for the mean squared error, regardless of whether slippage is considered or not, the simulation results are better than the experimental results when the weight of the error related to θ is the same. This indicates that the controller's performance in the simulation

[DOI: 10.1299/jamdsm.2025jamdsm0011]
© 2025 The Japan Society of Mechanical Engineers

is better than in the actual experiment. The differences between the experiment and the simulation may be attributed to several factors, including the inability of the controller to send commands at fixed time steps as in the simulation, assembly errors present in the robot prototype, and discrepancies between the linearized motion model used in the simulation and the actual motion model. Since we derived the linear kinematic equations of the robot using a first-order Taylor expansion at reference points on the reference path, they become imprecise when the robot's state deviates significantly from the reference value, making the controller less effective.

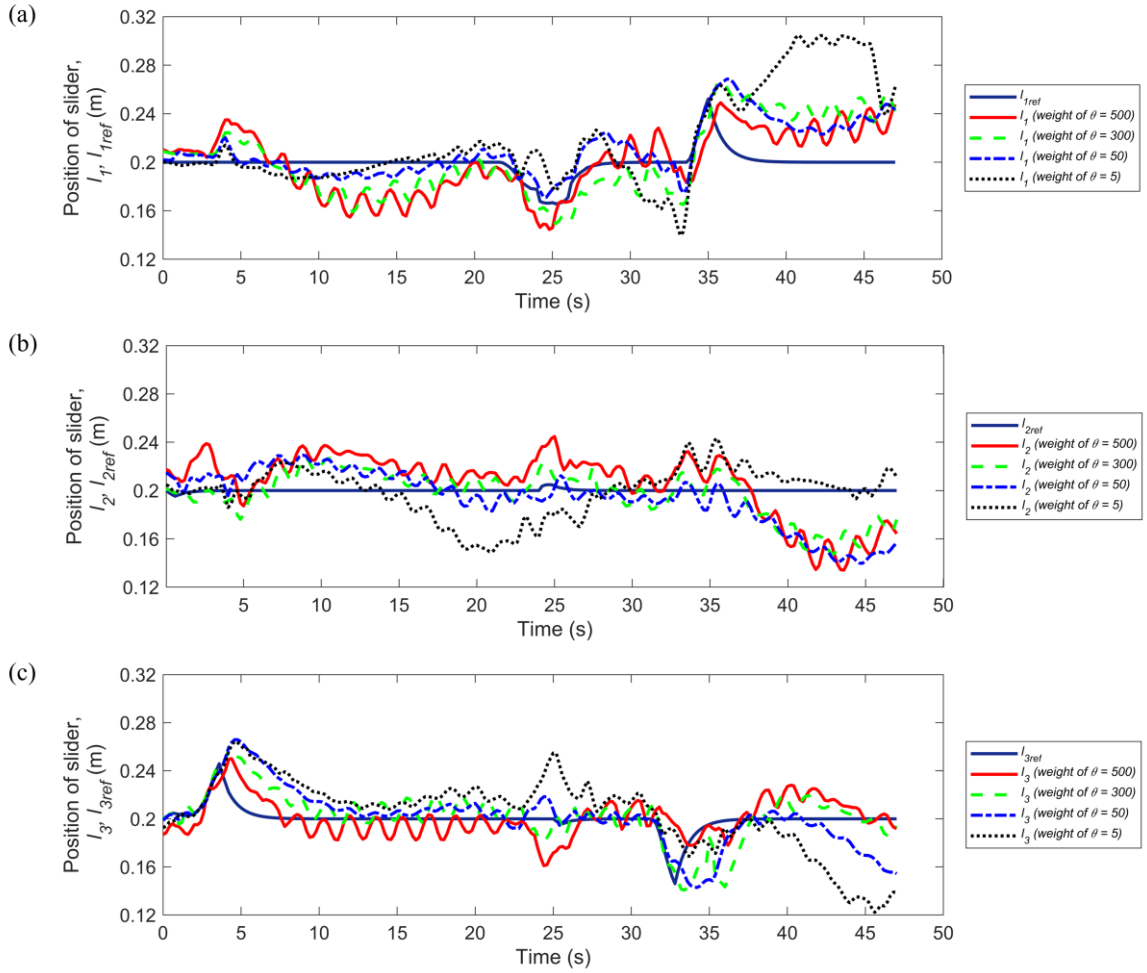


Fig. 10 Real and reference positions of sliders on SWOM in single-corner path experiment.

[DOI: 10.1299/jamdsm.2025jamdsm0011]
© 2025 The Japan Society of Mechanical Engineers

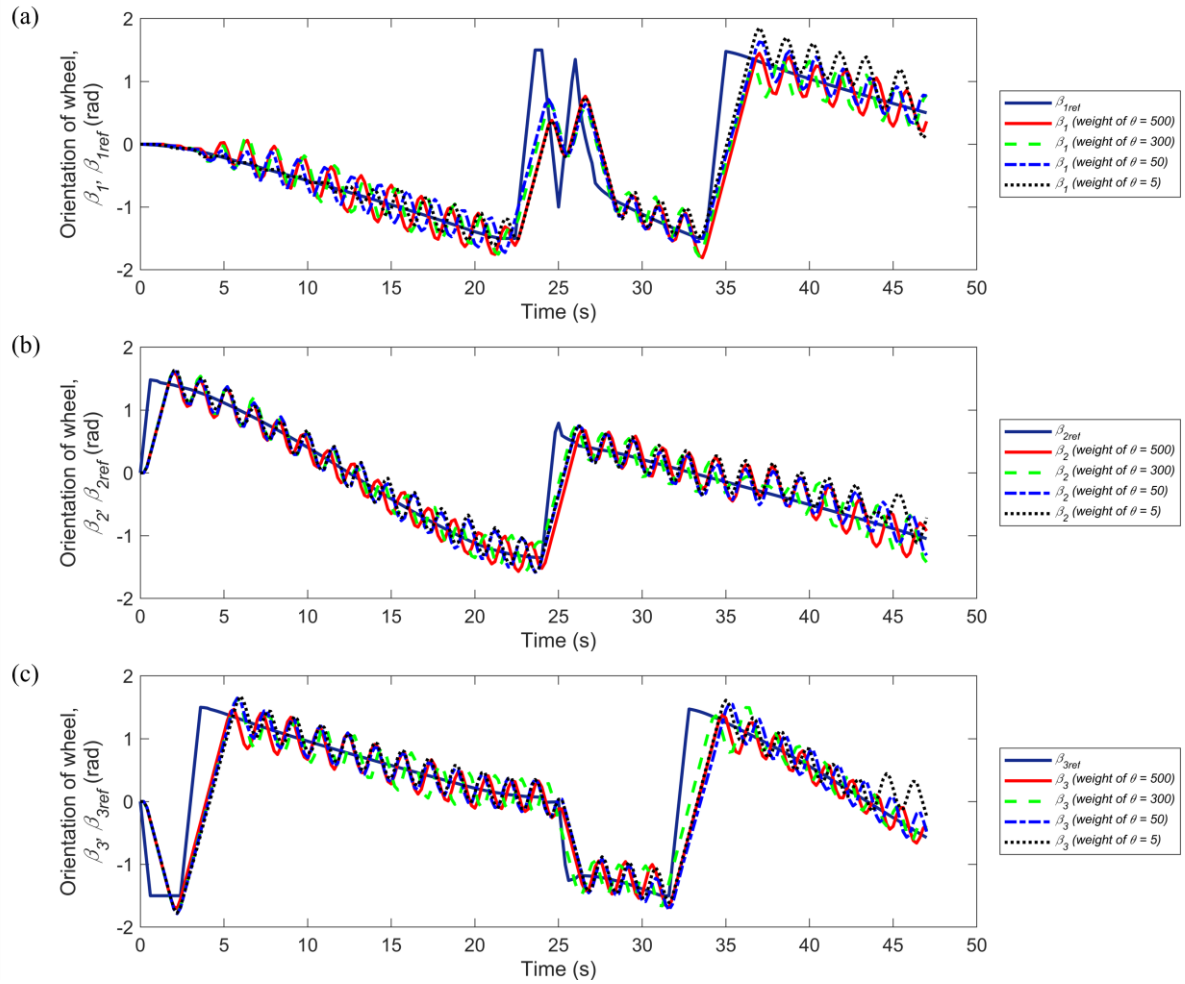


Fig. 11 Real and reference orientations of wheels on SWOM in single-corner path experiment.

The second reference path was smooth and circular. The robot needed to move along an arc with a radius of 0.5 m for 48 s at a speed of 0.03 m/s while rotating at 0.05 rad/s. The constraints on the inputs and outputs were the same as those for the single-corner path. The initial conditions were $X = 0.00$ m, $Y = 0.00$ m, $\theta = 0.00$ rad, and $\beta_1 = \beta_2 = \beta_3 = 0.00$ rad. The initial value of l_i was set randomly near the middle of the slide rail.

We also conducted three experiments and recorded the tracking errors from each experiment in Table 4. Comparing the data in Table 3, it can be observed that under the same parameter conditions, SWOM performs better in tracking the circular path than the single-corner path.

Table 4 Tracking error of each experiment (circular path).

		No. 1	No. 2	No. 3
Weight of θ = 50	Mean squared error	0.0012	0.0005	0.0007
	Max position error	0.0236	0.0262	0.0212
	Max angle error	0.1138	0.0662	0.0926

[DOI: 10.1299/jamdsm.2025jamdsm0011]
© 2025 The Japan Society of Mechanical Engineers

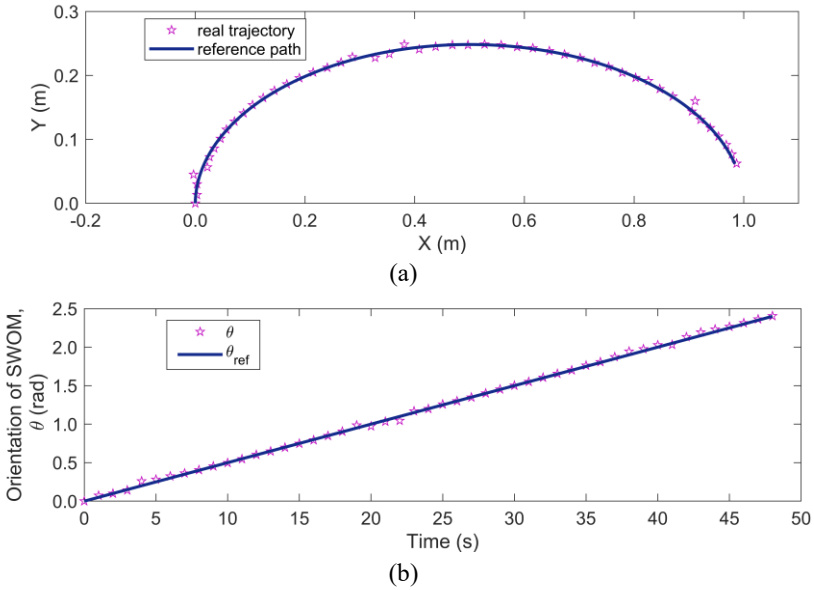


Fig. 12 (a) Real and reference paths and (b) real and reference orientations for SWOM in circular path experiment.

The following figures show the results in trial No. 2. Figure 12 shows the reference and actual paths of the robot as it tracked a circular path, along with the reference and actual orientations. Figure 13 shows the displacement of the sliders and the reference values. The results show that the robot worked properly in this situation. In Fig. 13, the peaks in the reference paths of the sliders are smoother than those observed for the single-corner path. This is because the robot moved along a circular arc while rotating. As a result, the robot's direction of motion changed more rapidly than it would moving in a straight line, preventing it from lingering too long in a direction parallel to the sliding rail.

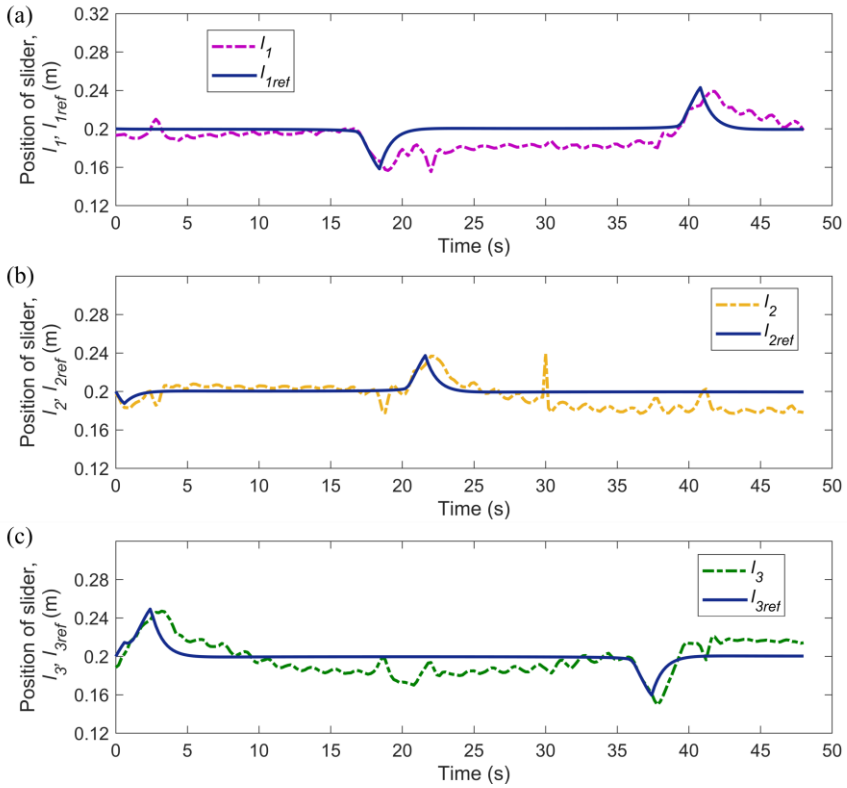


Fig. 13 Real and reference positions of sliders on SWOM in circular path experiment.

[DOI: 10.1299/jamdsm.2025jamdsm0011]
© 2025 The Japan Society of Mechanical Engineers

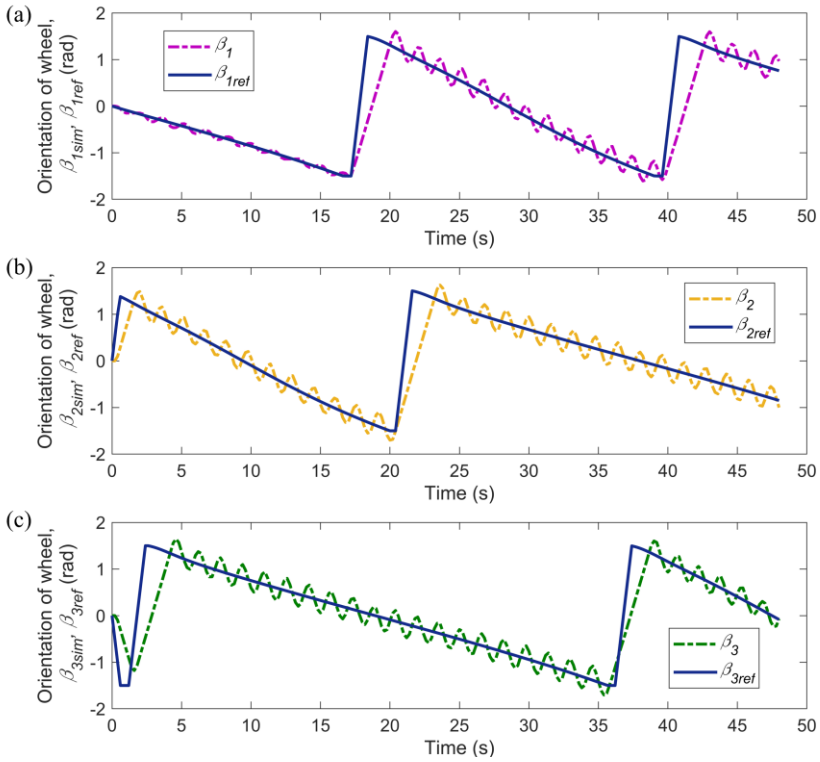


Fig. 14 Real and reference orientations of wheels on SWOM in circular path experiment.

Figure 14 shows the orientation of the robot's wheels during the experiment compared to the reference values. Similar to the single-corner path experiment, the experimental values oscillate around the reference values. This oscillation causes the wheels to rotate back and forth continuously, which can increase the likelihood of slippage. As explained in the discussion of the results from the single-corner path, if the position of β_i could be measured and set more accurately, this oscillation could be reduced.

The experiments using smooth and unsmooth reference paths confirm that the proposed controller works well, the tracking error is small, and the displacements of the sliders are controlled within the set limit.

6. Conclusion

In this paper, the LMPC method is applied to achieve trajectory tracking for SWOM. Building on the previous research, the linearized kinematic model of SWOM was obtained by performing a first-order Taylor expansion at reference points on the reference trajectory. We used only the reference trajectory to generate the linear kinematic model of SWOM and then used LMPC to compute the robot's inputs during subsequent control. To perform a Taylor expansion on the reference trajectory, the trajectory must include not only the displacement and orientation of the robot but also the displacement of the sliders, the orientation of the wheels, the wheels' rotation, and steering velocities. We proposed a simple and effective method for generating such a reference trajectory. By observing the generated reference trajectory, we found that when the target speed changes suddenly, the reference value of the sliders' displacement exhibits a peak. Similarly, when the target speed at the point on the robot that is in contact with an arbitrary slider is parallel to the corresponding sliding rail, the slider's displacement on that rail shows a peak. The peaks caused by sudden changes in speed can be mitigated by considering acceleration and deceleration. The peaks caused by the speed being parallel to the sliding rail can be reduced by increasing the rate of change in the robot's speed direction, such as by accelerating the robot's rotation or enabling it to move along a curve.

We conducted simulations considering wheel slippage and different weights for SOWM's orientation error in LMPC to validate the effectiveness of the proposed method. The results indicate that the SWOM's displacement, orientation, and wheel orientation are all able to effectively track the reference values. The displacement of the slider on the sliding rail is also kept within the allowable range. Experiments were also conducted on a prototype for both a single-corner path and a circular path. Due to the prototype's inability to precisely control the actual value of β, the

[DOI: 10.1299/jamdsm.2025jamdsm0011]
© 2025 The Japan Society of Mechanical Engineers

wheel orientation oscillated around the reference values and the slider position did not exactly match the reference values. Nevertheless, as long as the weight for SOWM's orientation error is not too small, the robot's displacement and orientation still tracked the reference values well, confirming the effectiveness of the proposed control method.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number JP21K14067.

References

Amoozgar, M.H., Sadati, S.H. and Alipour, K., Trajectory tracking of wheeled mobile robots using a kinematical fuzzy controller, International Journal of Robotics and Automation, Vol. 27, No.1 (2012), pp. 49-59.

Dragicevic, T. and Novak, M., Weighting factor design in model predictive control of power electronic converters: an artificial neural network approach, IEEE Transactions on Industrial Electronics, Vol. 66, No. 11 (2019), pp. 8870-8880.

Fujimoto, K., Sakurama, K. and Sugie, T., Trajectory tracking control of port-controlled Hamiltonian systems via generalized canonical transformations, Automatica, Vol. 39, No. 12 (2003), pp. 2059-2069.

He, W., Mu, X., Zhang, L. and Zou, Y., Modeling and trajectory tracking control for flapping-wing micro aerial vehicles, IEEE/CAA Journal of Automatica Sinica, Vol. 8, No.1 (2021), pp.148-156.

Kato, K. and Wada, M., ACROBAT-S omnidirectional mobile robot prototype: and study on ball drive mechanism, Proceedings of 2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM) (2020), pp. 105-110.

Lin, G., Terakawa, T., Shinno, K., Inoue, T. and Komori, M., RoMop: a new type of wheeled mobile platform based on rotating locomotion, IEEE/ASME Transactions on Mechatronics, Vol. 29, No. 4 (2024), pp. 2510-2521.

Liu, Y. K. and Zhang, Y. M., Model-based predictive control of weld penetration in gas tungsten arc welding, IEEE Transactions on Control Systems Technology, Vol. 22, No.3 (2014), pp. 955-966.

Liu, Y., Li, H., Ding, L., Liu, L., Liu, T., Wang, J. and Gao, H., An omnidirectional mobile operating robot based on mecanum wheel, Proceedings of 2017 2nd International Conference on Advanced Robotics and Mechatronics (ICARM) (2017), pp. 468-473.

Li, Z., Deng, J., Lu, R., Xu, Y., Bai, J. and Su, C.Y., Trajectory-tracking control of mobile robot systems incorporating neural-dynamic optimized model predictive approach, IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 46, No. 6 (2016), pp. 740-749.

Mayne, D.Q., Model predictive control: Recent developments and future promise, Automatica, Vol. 50, No. 12 (2014), pp. 2967-2986.

Medina, O. and Hacohen, S., Overcoming kinematic singularities for motion control in a caster wheeled omnidirectional robot, Robotics, Vol. 10, No. 4 (2021), DOI: 10.3390/robotics10040133.

Navabi, H., Sadeghnejad, S., Ramezani, S. and Baltes, J., Position control of the single spherical wheel mobile robot by using the fuzzy sliding mode controller, Advances in Fuzzy systems (2017), p. 2651976, DOI: 10.1155/2017/2651976.

Neunert, M., Crousaz, C. de, Furrer, F., Kamel, M., Farshidian, F., Siegwart, R. and Buchli, J., Fast nonlinear model predictive control for unified trajectory optimization and tracking, Proceedings of 2016 IEEE International Conference on Robotics and Automation (ICRA) (2016), pp. 1398-1404.

Qiao, L. and Zhang, W., Double-loop integral terminal sliding mode tracking control for UUVs with adaptive dynamic compensation of uncertainties and disturbances, IEEE Journal of Oceanic Engineering, Vol. 44, No. 1 (2019), pp. 29-53.

Qiao, L. and Zhang, W., Trajectory tracking control of AUVs via adaptive fast nonsingular integral terminal sliding mode control, IEEE Transactions on Industrial Informatics, Vol. 16, No.2 (2020), pp. 1248-1258.

Shen, C., Shi, Y. and Buckham, B., Trajectory tracking control of an autonomous underwater vehicle using lyapunov-based model predictive control, IEEE Transactions on Industrial Electronics, Vol. 65, No. 7 (2018), pp. 5796-5805.

Steyn, C. W. and Sandrock, C., Benefits of optimisation and model predictive control on a fully autogenous mill with

 © 2025 The Japan Society of Mechanical Engineers

variable speed, Minerals Engineering, Vol. 53 (2013), pp. 113-123.

Taheri, H. and Zhao, C. X., Omnidirectional mobile robots, mechanisms and navigation approaches, Mechanism and Machine Theory, Vol. 153 (2020), DOI: 10.1016/j.mechmachtheory.2020.103958.

Terakawa, T., Komori, M. and Fujimoto, K., Control of an omnidirectional mobile robot with wheels connected by passive sliding joints, Journal of Advanced Mechanical Design, Systems, and Manufacturing, Vol. 13, No. 1 (2019), DOI: 10.1299/jamdsm.2019jamdsm0006.

Terakawa, T., Komori, M., Matsuda, K. and Mikami, S., A novel omnidirectional mobile robot with wheels connected by passive sliding joints, IEEE/ASME Transactions on Mechatronics, Vol. 23, No. 4 (2018), pp. 1716-1727.

Terakawa, T., Yogou, M. and Komori, M., Motion characteristics analysis of a mecanum-wheeled omnidirectional mobile robot on a slope, Proceedings of IFToMM World Congress on Mechanism and Machine Science, Vol. 148 (2023), pp. 733-741.

Wang, C., Liu, X., Yang, X., Hu, F., Jiang, A. and Yang, C., Trajectory tracking of an omni-directional wheeled mobile robot using a model predictive control strategy, Applied Sciences, Vol. 8, No. 2 (2018),DOI: 10.3390/app8020231.

Wang, K., Liu, Y. and Li, L., Visual servoing trajectory tracking of nonholonomic mobile robots without direct position measurement, IEEE Transactions on Robotics, Vol. 30. No. 4 (2014), pp. 1026-1035.

Wehr, M., Schätzler, S., Abel, D. and Hirt, G., Model predictive control of an overactuated roll gap with a moving manipulated variable, Proceedings of 2020 American Control Conference (ACC) (2020), pp. 1931-1936.

Wendemagegn, Y.A., Asfaw, W.A., Abdissa, C.M. and Lemma, L.N., Enhancing trajectory tracking accuracy in three-wheeled mobile robots using backstepping fuzzy sliding mode control, Engineering Research Express, Vol, 6, No. 4 (2024), DOI: 10.1088/2631-8695/ad79b9.

Yu, R., Zhao, H., Zhen, S., Huang, K., Chen, X., Sun, H. and Shao, K., A novel trajectory tracking control of AGV based on Udwadia-Kalaba approach, IEEE/CAA Journal of Automatica Sinica, Vol. 11, No. 4 (2024), pp. 1069-1071.

Zarei, F., Moosavian, S. and Najafi, A., Force analysis of a spherical single-wheeled robot, Proceedings of the 6th RSI International Conference on Robotics and Mechatronics (IcRoM) (2018), pp. 542-547.

[DOI: 10.1299/jamdsm.2025jamdsm0011]
© 2025 The Japan Society of Mechanical Engineers